

DESCRIPTION

OPTICAL DISC, RECORDING APPARATUS, READING APPARATUS, AND METHOD

5 Technical Field

[0001]

The present invention relates to a technical field including a technology for temporarily recording defect entries.

10 Background Art

[0002]

The defect entry is information that correlates positions of defective clusters with positions of substitute sectors that are used in place of the defective clusters.

15 The temporarily recording of the defect entry is a process in which when a defective area is detected while data is recorded onto a write-once optical disc, a defect entry concerning the defective area is recorded into a temporary area. The reason why it is necessary to temporarily write the defect entry is as follows. In the optical disc, an area called Defect Management  
20 Area (DMA) for writing defect entries therein is preliminarily provided. However, it is not allowed in a write-once optical disc to write a defect entry into the DMA until the finalize process is performed. This is because once data is written onto a  
25 write-once optical disc, it is impossible to rewrite the data later. That is to say, once a defect entry concerning a defective area is written to the DMA, it is impossible to add or update data in the DMA even if another defective area is detected thereafter. For this reason, when a defective area is detected

during a recording process, it is necessary to write a defect entry concerning the defective area into a temporary defect management area, or the Temporary Disc Management Structure (TDMS).

5 [0003]

The document indicated below discloses, as a prior art, a technology for writing the temporary defective area list, or the Temporary DeFect List (hereinafter, TDFL) that includes a plurality of defect entries concerning defective areas.

10 Document 1: Japanese Patent No. 2671656 (Figures 3-6)

#### Disclosure of the Invention

#### The Problems the Invention Is Going to Solve

[0004]

15 In the DMA, a plurality of defective area lists are arranged such that the addresses of defective areas indicated by the defective area lists are arranged sequentially in the ascending order. With such arrangement, if the data is read optically from the optical disc in order from the inner circumference to the  
20 outer circumference, the plurality of defective area lists are read out onto the memory in the ascending order of the addresses of the defective areas indicated by the defective area lists. The firmware of the drive apparatus performs a process using the defective area lists on presumption that the defective area lists  
25 are arranged in the above-mentioned order on the memory. The firmware of the drive apparatus requires the defective area lists to be provided in the above-mentioned order since it is primitive software for controlling what is called the physical layer.

[0005]

The defective area lists recorded in the TDMS have the same properties as the defective area lists recorded in the DMA, and therefore need to be read and arranged onto the memory in the ascending order of the addresses of the defective areas indicated by the defective area lists. Otherwise, the firmware would fail to process the defective area lists normally.

For the reasons stated above, the defective area lists recorded in the TDMS of a conventional optical disc are arranged sequentially in the ascending order of the addresses of the defective areas indicated by the defective area lists. However, this requirement of arranging the defective area lists in the ascending order of the addresses of the defective areas ruins the freedom in arranging the defective area lists in conventional optical discs. With such a lack of freedom, if a defective cluster is detected in the middle of recording a plurality of defective area lists, the plurality of defective area lists cannot be recorded discretely, but all of them must be re-recorded in the clusters succeeding the defective cluster.

[0006]

As described above, since all the defective area lists should be re-recorded in clusters after a defective cluster, the areas allocated to the TDMS are consumed rapidly.

Fig. 1 shows how the process is carried out when a defective cluster is detected while a plurality of defective area lists are written. The first row of Fig. 1 represents a plurality of clusters that constitute TDMS#j. The drawing shows a plurality of temporary defective area lists. Among the clusters constituting the TDMS in the first row, cluster#2 is a defective cluster. The second row indicates the state in which TDFL#1 has

been written in cluster#1, and the third row indicates the state in which TDFL#2 failed to be written into cluster#2. According to a conventional technology, TDFL#1 to TDFL#4 should be arranged sequentially. As a result, TDFL#1 to TDFL#4 and Temporary Disc  
5 Definition Structure (TDDS) are written in cluster#3 and the succeeding clusters as a retry. The TDDS is information that indicates the position of the first TDFL (address C3 in Fig. 1) among TDFL#1 to TDFL#4. The fourth to seventh rows show processes in which TDFL#1 to TDFL#4 and TDDS are written in cluster#3 and  
10 the succeeding clusters.  
[0007]

Here, the TDMS is assigned to a specific area called the lead-in area, on the optical disc. Accordingly, such a conventional arrangement of TDFLs, which requires that if a  
15 defective cluster is detected, all of a set of defective area lists should be re-recorded in the clusters after the defective cluster, has a problem that the areas allocated to the TDMS are consumed rapidly.

It is therefore an object of the present invention to provide  
20 an optical disc that enables a plurality of defective area lists to be read out onto the memory in a predetermined order, but prevents the temporary defect management area (TDMS) from being consumed rapidly.

25 Means to Solve the Problems  
[0008]

The above-mentioned object of the present invention is achieved by a write-once optical disc that includes a temporary defect management area, wherein a plurality of defective area

lists and structure information are recorded in the temporary defect management area, the plurality of defective area lists indicate at least one defective area in the optical disc, the structure information includes a plurality of pieces of position information that indicate positions of the defective area lists in the temporary defect management area, the plurality of pieces of position information corresponding one-to-one to the defective area lists, and the plurality of pieces of position information are arranged in an order in which the defective area lists corresponding thereto are read out.

#### Effects of the Invention

[0009]

With the above-described construction of the optical disc of the present invention, even if the defective area lists are arranged discretely in the temporary defect management area, the defective area lists are read out onto the memory in an order in which the plurality of pieces of position information corresponding thereto are arranged in the structure information, so that the defective area lists are read out in a predetermined order and are arranged in the memory in the same order.

[0010]

When the predetermined order is an ascending order of the addresses of defective areas indicated by the defective area lists, the defective area lists are read out and arranged on the memory in the ascending order of the addresses of defective areas indicated by the defective area lists.

Since the order of reading out the defective area lists is indicated by the plurality of pieces of position information

that respectively correspond to the defective area lists, there is no need to arrange the defective area lists sequentially even if the temporary defect management area includes a defective cluster. The defective area lists can be arranged in discrete sequential areas including the defective cluster. That is to say, a random arrangement of the defective area lists is possible.

[0011]

With the above-stated construction, the defective area lists can be arranged randomly even if the temporary defect management area includes a defective cluster. Accordingly, with the above-stated construction, when a plurality of defective area lists are to be written in a sequence of clusters including a defective cluster, only part of the defective area lists needs to be written as a retry, instead of all of the defective area lists. As understood from this, the construction of the present invention prevents the temporary defect management area (TDMS) from being consumed rapidly since it reduces the defective area lists to be written in a retry.

[0012]

With such a construction in which defective area lists need not to be written sequentially in order into a defective cluster and the succeeding clusters, the freedom in arranging the defective area lists is increased.

## Brief Description of the Drawing

[0013]

Fig. 1 shows how the process is carried out when a defective cluster is detected while a set of defective area lists is written.

Fig. 2A shows the overall structure of the optical disc.

Fig. 2B shows the internal structure of the cluster.

Fig. 3 shows tracks in the state where they are horizontally extended though they are in reality formed spirally on the optical disc 1.

5            Fig. 4A shows the internal structure of the lead-in area  
2.

[0014]

Fig. 4B shows the internal structure of the lead-out area  
4.

10           Fig. 5A shows a common structure of the 1<sup>st</sup> DMA to the 4<sup>th</sup>  
DMA.

Fig. 5B shows the data structure of the DFL.

Fig. 5C shows the data structure of the DDS.

Fig. 6 shows the structure of a TDMA.

15           Fig. 7A shows the data structure of the TDFL.

[0015]

Fig. 7B shows the data structure of the TDDS.

Fig. 8 shows one example of a discrete arrangement of TDFLs.

Fig. 9 shows the internal structure of the recording/reading  
20 apparatus 100.

Fig. 10 shows the internal structure of the defect management  
information processing unit 13.

Figs. 11A to 11D show reading of TDMS information by the  
TDMS information reading unit 22 and writing of TDMS information  
25 by the TDMS information writing unit 26.

Figs. 12A to 12C show writing of DMA information by the  
DMA information writing unit 29.

Fig. 13 is a flowchart showing the procedure of the DFL  
read process.

Fig. 14 is a flowchart showing the procedure of the read/write process to be performed in accordance with a command issued from the high-order control apparatus 200.

Fig. 15 is a flowchart showing the procedure of the  
5 write-to-TDMS process.

Fig. 16 shows how the recording/reading apparatus 100 of the first embodiment writes the TDFLs.

Fig. 17 shows an example of a discrete arrangement of TDFLs in the second embodiment.

10 Fig. 18 is a flowchart showing the procedure of the write-to-TDMS process in the second embodiment.

Fig. 19 shows the process in which TDFLs are written by the recording/reading apparatus 100 of the second embodiment.

15 Fig. 20 shows an example of a discrete arrangement of TDFLs in the third embodiment.

Fig. 21 is a flowchart showing the procedure of the write-to-TDMS process in the third embodiment.

Fig. 22 shows how the recording/reading apparatus 100 of the third embodiment writes the TDFLs.

20

#### Description of Characters

[0016]

- 1 optical disc
- 2 lead-in area
- 25 3 data area
- 4 lead-out area
- 5 spare area
- 6 user data area
- 7 spare area



- 11 BD-R drive
- 12 instruction processing unit
- 13 defect management information processing unit
- 14 reproduction control unit
- 5 15 recording control unit
- 16 storage buffer
- 21 DMA information reading unit
- 22 TDMS information reading unit
- 23 control memory
- 10 24 defect entry adding unit
- 25 TDFL conversion unit
- 26 TDMS information writing unit
- 27 verifying unit
- 28 position information generating unit
- 15 29 DMA information writing unit
- 100 recording/reading apparatus
- 200 high-order control apparatus

#### Best Mode for Carrying Out the Invention

20 [0017]

Here will be described an embodiment of the optical disc of the present invention. Fig. 2A shows the overall structure of the optical disc. Optical disc 1 is a large-capacity BD-R for which blue laser is used, and has a large number of tracks  
 25 formed spirally thereon. Each track is divided into a plurality of clusters.

Fig. 2B shows the internal structure of the cluster. Each cluster is a set of sectors for error correction, and is also called an error correction block. The error correction is made

in units of clusters. Accordingly, recording or reproduction is performed using the cluster as the minimum unit. As shown in Fig. 2B, each cluster is composed of 32 sectors, and is 64KB in size.

5 [0018]

Fig. 3 shows tracks in the state where they are horizontally extended though they are in reality formed spirally on the optical disc 1. The third row of Fig. 3 shows the horizontally extended tracks, and the second row shows a lead-in area 2, a data area 10 3, and a lead-out area 4 that are formed on the tracks.

The lead-in area 2 stores control information referred to by an apparatus, and plays a role of "supplementary area" so that if the optical head overruns when trying to access a track at the edge of the data area 3, the optical head can follow the track.

15 [0019]

The data area 3 is a substantial portion of the optical disc.

The lead-out area 4 stores control information referred to by an apparatus, and plays a role of "supplementary area" so 20 that if the optical head, trying to access a track at the edge of the data area 3, overruns, it can follow the track.

The first row of Fig. 3 shows the internal structure of the data area 3. As shown in the first row, the data area 3 includes two spare areas, 5 and 7, and a user data area 6.

25 [0020]

The user data area 6 is an area into which the user can store information he/she likes to, including real-time data such as music or video, text, and computer data such as a database.

The spare areas 5 and 7 are substitute areas which are used

to record data in place of a defective cluster that may be detected in the user data area 6.

#### <Internal Structure of Lead-in Area 2>

Fig. 4A shows the internal structure of the lead-in area 2. The second row of Fig. 4A shows the lead-in area 2 as a whole, and the first row shows the internal structure of the lead-in area 2. As shown in the first row, the lead-in area 2 includes the First Defect Management Area (hereinafter 1<sup>st</sup> DMA), Second Defect Management Area (hereinafter 2<sup>nd</sup> DMA), and Temporary Defect Management Area (hereinafter TDMA). The 1<sup>st</sup> DMA and the 2<sup>nd</sup> DMA are areas used for managing the information of defective clusters in the optical disc 1.

[0021]

#### <Internal Structure of Lead-out Area 4>

Fig. 4B shows the internal structure of the lead-out area 4. The second row of Fig. 4B shows the lead-out area 4 as a whole, and the first row shows the internal structure of the lead-out area 4. As shown in the first row, the lead-out area 4 includes Third Defect Management Area (hereinafter 3<sup>rd</sup> DMA) and Fourth Defect Management Area (hereinafter 4<sup>th</sup> DMA). The 3<sup>rd</sup> DMA and the 4<sup>th</sup> DMA are areas used for managing the information of defective clusters in the optical disc 1.

[0022]

The 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA are respectively disposed at predetermined positions, and are variable in size, varying depending on the number of defective clusters.

The 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA have a common structure, as shown in Fig. 5A. Each DMA includes Disc Definition Structure (hereinafter, DDS) and a defective area list, or DeFect List

(hereinafter, DFL). They are arranged such that the DDS is followed by the DFL.

[0023]

#### <Data Structure of DFL>

5 Fig. 5B shows the data structure of the DFL.

The DFL includes a defect list header and defect entry #1 to defect entry #M, where the number of the defect entries included therein ranges from 0 to M.

10 The defect list header contains information such as the number of defect entries included in the DFL.

[0024]

Each of the defect entries #1 to #M includes "defective cluster start position information" which shows the start position of a defective cluster detected during an access to the user data area 6, and also includes "substitute cluster position information" which shows the position of a cluster in the spare area 5 or 7 used in place of the defective cluster.

#### <Data Structure of DDS>

Fig. 5C shows the data structure of the DDS. The DDS includes "DFL start position information (DFL pointer)" which shows the position of the first DFL among one or more DFLs in each DMA, and also includes "other information". Since the DDS indicates the DFL starting position, it is possible to read the DFLs in each DMA by accessing the DDS.

25 [0025]

The 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA need to contain the same information, except for the starting position information of the DFL. Each of the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA is composed of 32 clusters, among which the first four clusters are used for DDS, and four DDSs

are recorded therein repeatedly. The succeeding 28 clusters are used for DFL, and are used in units of four clusters. More specifically, first the four clusters 5-8 are used, then if they become unassured, the next four clusters 9-12 are used, and so on. Up to now, DFL and DDS have been described.

[0026]

<TDMA>

Here, TDMA will be described. The TDMA is an area for temporarily recording defect entries that have been generated before the finalize process is performed. The TDMA is unique to write-once optical discs, and is not provided in the read-only optical discs (BD-ROM) or rewritable optical discs (BD-RE).

[0027]

The finalize process is a process in which a write-once optical disc is modified in data structure so as to be compatible with a rewritable optical disc (hereinafter, performing the finalize process for an optical disc may also be referred to as finalizing the optical disc).

The technical significance of having the TDMA in the lead-in area 5 is as follows. In the case of a rewritable optical disc, the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA can be rewritten, and it is possible to write, into the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA, a defect entry concerning a defective area that was detected most recently, and rewrite the defect entry repeatedly as necessary. It is therefore possible to cause the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA to have information concerning the defective area that was detected most recently.

[0028]

However, in the case of a write-once optical disc, data can be recorded into the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA only once. It is

therefore impossible for the write-once optical disc to cause the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA to have information concerning the defective area that was detected most recently. Accordingly, write-once optical discs are provided with the TDMA so as to be  
5 compatible with rewritable optical discs.

#### <Internal Structure of TDMA>

Fig. 6 shows the structure of a TDMA. The second row of Fig. 6 shows the internal structure of the TDMA. As shown in the second row, the TDMA includes the N number of TDMSs (TDMS#1, TDMS#2, TDMS#3, . . . TDMS#N). The N number of TDMSs are areas into which information (defect entries), which concerns defective areas that are detected while data is recorded into the user data area, is written such that a defect entry concerning a defective area detected firstly is written in the first TDMS (TDMS#1), a  
10 defect entry concerning a defective area detected secondly is written in the second TDMS (TDMS#2), . . . a defect entry concerning a defective area detected Nthly is written in the N<sup>th</sup> TDMS (TDMS#N).  
[0029]

The later written a defect entry is, the larger number a TDMS has, and only a TDMS with the largest number is made valid (hereinafter, the valid TDMS may also be referred to as the latest TDMS). When the optical disc 1 is accessed, only the TDMS with the largest number among those having defect entries is referred to. In the finalize process, only the defect entry written in  
20 the latest TDMS is written into the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA.

The TDMSs have a common structure. That is to say, as shown in the first row of Fig. 6, each TDMS includes one or more temporary defective area lists, or one or more Temporary DeFect Lists (hereinafter, TDFLs) and Temporary Disc Definition Structure

(hereinafter, TDDS). They are arranged such that the TDFLs are followed by the TDDS.

[0030]

The TDMS is structured in units of clusters, and is variable  
5 in size, varying depending on the number of defective entries.  
Each TDMS is thus composed of one or more clusters. When a TDMS  
is composed of one cluster, a TDFL and a TDDS are included in  
the cluster. When a TDMS is composed of a plurality of clusters,  
a TDFL and a TDDS are stored in a cluster at the end of the TDMS,  
10 and only TDFLs are stored in the other clusters.

[0031]

#### <Data Structure of DFL>

Fig. 7A shows the data structure of the TDFL.

The TDFL has the same structure as the DFL. As indicated  
15 by the lead fl, the TDFL includes a defect list header and defect  
entry #1 to defect entry #M, where the number of the defect entries  
included therein ranges from 0 to M.

The defect list header contains such information as the  
number of defect entries included in the TDFL.

20 [0032]

Each of the defect entries #1 to #M includes "defective  
cluster position information" which shows the position of a  
defective cluster detected during an access to the user data area  
6, and also includes "substitute cluster-position information"  
25 which shows the position of a cluster in a spare area 5 or 7 used  
in place of the defective cluster.

#### <Data Structure of TDDS>

Fig. 7B shows the data structure of the TDDS. The TDDS is  
information having a fixed size, and is composed of one sector

that is 2KB in size. The TDDS has the same size as the DDS. The TDDS is assigned to a sector at a predetermined position among the 32 sectors constituting the last cluster of the TDMS. For example, the TDDS is assigned to the last sector among the 32 sectors constituting the last cluster of the TDMS. As shown in Fig. 7B, the TDDS includes a plurality of pieces of TDFL position information (TDFL pointers) which show starting positions of TDFLs, and also includes "other information". The TDDS, as is the case with the DDS, indicates positions of DFLs, but differs critically from the DDS in that when the TDMS includes a plurality of TDFLs, the TDDS includes a plurality of pieces of TDFL position information that respectively show the starting positions of the TDFLs. The plurality of pieces of TDFL position information are arranged in the ascending order of the addresses of the defective areas that are indicated by the TDFLs which correspond one-to-one to the plurality of pieces of TDFL position information. And therefore, the TDFL#1 position information is adjacent to the TDFL#2 position information in the TDDS. When there are four TDFLs, TDFL#1 to TDFL#4, in one TDMS, the positions of the four TDFLs are indicated by the TDDS. Since the TDDS indicates positions of the TDFLs, it is possible to arrange the TDFLs discretely in one TDMS.

[0033]

The demand for arranging the TDFLs discretely is especially strong for the case where a set of clusters constituting one TDMS includes a defective cluster.

#### <Discrete Arrangement of TDFLs>

Fig. 8 shows one example of a discrete arrangement of TDFLs. The first row of Fig. 8 shows a plurality of clusters (cluster#1,



cluster#2, cluster#3, . . . cluster#7) that constitute a TDMS. The second row shows four TDFLs (TDFL#1, TDFL#2, TDFL#3 and TDFL#4) and a TDDS, which are written in these clusters.

[0034]

5           The TDMS#j shown in Fig. 8 is composed of cluster#1 to cluster#7, among which cluster#2 is a defective cluster. In Fig. 8, TDFL#1 to TDFL#4 are written in cluster#1 which precedes the defective cluster#2, and in cluster#3 to cluster#5 which succeed the defective cluster#2, which means TDFL#1 to TDFL#4 are written  
10   into discontinuous clusters.

          The TDDS shown in Fig. 8 includes TDFL#1 position information (TDFL#1: address C1) and TDFL#2 position information (TDFL#2: address C3), where the TDFL#1 position information indicates the address C1, which is the address of the last cluster (cluster#1)  
15   among clusters that precede the defective cluster, as the start position of TDFL#1, and the TDFL#2 position information indicates the address C3, which is the address of the first cluster (cluster#3) among clusters (cluster#3 to cluster#5) that succeed the defective cluster, as the start position of TDFL#2. The TDDS  
20   shown in Fig. 8 is characterized in that a plurality of pieces of TDFL position information (TDFL#1 position information, TDFL#2 position information, TDFL#3 position information, and TDFL#4 position information) are arranged in the ascending order of the addresses of the defective areas indicated by the corresponding  
25   TDFLs, and that the TDFL#1 position information is adjacent to the TDFL#2 position information in the TDDS. With such a construction, if the TDFLs are read in accordance with the position information in the TDDS, TDFL#1 to TDFL#4 are read in a predetermined order regardless of whether there is a defective

cluster, and they can be arranged in the memory.

[0035]

As described above, the TDDS in the present embodiment includes four pieces of TDFL position information which respectively indicate the start positions of the clusters in which  
5 corresponding four TDFLs are written. This enables TDFL#1 and TDFL#2 to TDFL#4 to be arranged discretely. For this reason, TDFL#2 to TDFL#4, which would be recorded, starting with TDFL#1, in a series of clusters succeeding the defective cluster#2 in  
10 the TDMS according to conventional technologies, are recorded in cluster#3 to cluster#5 immediately after the defective cluster#2 in the TDMS.

[0036]

With such a construction in which the TDDS indicates the  
15 positions (addresses C1, C3, C4, and C5) of the clusters in which TDFL#1 to TDFL#4 are respectively recorded, even if TDFL#1 to TDFL#4 are arranged discretely, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the apparatus for reproduction, only  
20 by accessing the clusters in the TDMS in accordance with the positions indicated by the TDDS.

It would be needless to say that although in the present embodiment, the second cluster in the TDMS is a defective cluster, it is an example case provided for the sake of convenience, and  
25 there is no need for the second cluster to be defective, and there may be any number of defective clusters.

[0037]

As apparent from the above description, in the optical disc 1 in the first embodiment, a plurality of pieces of TDFL position

information are stored in the TDDS in the ascending order of defect entries, and the TDFLs can be recorded in discrete areas excluding the defective areas, enabling the TDMA, which has a finite size, to be used efficiently.

5 <Comparison between DFL Position information and TDFL Position Information>

DFLs are recorded in a plurality of discrete areas (1<sup>st</sup> DMA to 4<sup>th</sup> DMA) on the optical disc 1. A plurality of pieces of DFL position information are respectively stored in areas at predetermined positions on the optical disc 1 (a piece of DFL position information is stored in the DDS which is at the start of each of 1<sup>st</sup> DMA to 4<sup>th</sup> DMA), in the ascending order of the addresses of the defective areas indicated by the defect entries.

10 [0038]

TDFLs are recorded in a plurality of discrete areas (areas before and after the defective cluster shown in Fig. 8) on the optical disc 1. A plurality of pieces of TDFL position information are stored in an area at a predetermined position on the optical disc 1 (a plurality of pieces of TDFL position information are stored in the TDDS which is positioned immediately before not-recorded areas in the TDMS), in the ascending order of the addresses of the defective areas indicated by the defect entries.

15 20

It can be said that the DFL recording method and the TDFL recording method have in common that position information, which indicates a position of a defect entry, is recorded in a predetermined area. As a result, the reproduction apparatus can use both DFLs and TDFLs following a common procedure regardless of whether the defective entries are included in DFLs or TDFLs, namely, by reading a plurality of pieces of position information from a predetermined area on the optical disc 1 (DDSs positioned

25

at the start of each of 1<sup>st</sup> DMA to 4<sup>th</sup> DMA, or TDDS positioned immediately before not-recorded areas in the TDMS), and reading the defective entries according to the plurality of pieces of position information in an order thereof.

5 [0039]

As explained above, the present embodiment achieves a control procedure using TDFLs that is compatible with a control procedure using DFLs. This enables such a control procedure using TDFLs to be implemented in the reproduction apparatus, without  
10 forcing the reproduction apparatus to use complex software for the control. As such, the TDFL position information of the present embodiment produces excellent advantageous effects that it achieves compatibility with the procedure using DFLs and simplifies the control software in the reproduction apparatus.

15 [0040]

This completes the description of an embodiment of the optical disc of the present invention.

#### <Recording/Reading Apparatus 100>

From now on, an embodiment of the recording/reading  
20 apparatus of the present invention will be described. A recording/reading apparatus 100 shown in Fig. 9 is an apparatus that is provided with the functions of both the recording and reading apparatuses of the present invention. The following explanation will be provided in regards with the recording/reading  
25 apparatus 100.

[0041]

In Fig. 9, the recording/reading apparatus 100 is connected to a high-order control apparatus 200 via an I/O bus. Typically, the high-order control apparatus 200 is a host computer. In

accordance with the commands issued by the high-order control apparatus 200, reading data from or writing data to the optical disc 1 are executed. In this writing, no data is recorded in the DMA until the finalize process is performed, and information  
5 concerning one or more defective clusters detected during recording is written in the TDMS.

[0042]

In Fig. 9, the recording/reading apparatus 100 includes a drive mechanism 11, an instruction processing unit 12, a defect  
10 management information processing unit 13, a reproduction control unit 14, a recording control unit 15, and a storage buffer 16.

The drive mechanism 11 performs loading/ejecting of the optical disc 1, optically reads the contents of the optical disc 1 from it, and optically writes data onto the optical disc 1.

15 [0043]

The instruction processing unit 12 reads or writes user data from/to the optical disc 1 in accordance with a command issued from the high-order control apparatus 200.

The defect management information processing unit 13  
20 accesses the optical disc 1 using the defect entries, and updates the TDMS.

[0044]

It should be noted here that "accessing the optical disc 1 using the defect entries" means that if a read or write destination  
25 specified by the command from the high-order control apparatus 200 is a defective cluster indicated in a defect entry, a substitute cluster is accessed in place of the defective cluster. Also, if a defective area is newly detected during an access to the user data area of the optical disc 1 by the instruction processing

unit 12, a defect entry concerning the newly detected defective area is newly generated.

[0045]

It should also be noted that "updating the TDMS" means that  
5 in recording of data, the defect entry concerning the newly detected defective area is added to the TDFL in the latest TDMS, and writes the TDFL, to which the defect entry concerning the newly detected defective area was added, into the next TDMS.

In accordance with an instruction from the instruction  
10 processing unit 12 or the defect management information processing unit 13, the reproduction control unit 14 controls the drive mechanism 11 to read data from a desired cluster on the optical disc 1.

[0046]

15 In accordance with an instruction from the instruction processing unit 12 or the defect management information processing unit 13, the recording control unit 15 controls the drive mechanism 11 to write data into a desired cluster on the optical disc 1.

The storage buffer 16 is a buffer used to temporarily store  
20 information read from DMA (DMA information) and information read from TDMS (TDMS information).

[0047]

Up to now, the internal structure of the recording/reading apparatus 100 has been described.

## 25 <Internal Structure of Defect Management Information Processing Unit 13>

Here, the internal structure of the defect management information processing unit 13 will be described in detail. Fig. 10 shows the internal structure of the defect management information processing unit 13. In Fig. 10, the defect management

information processing unit 13 includes a DMA information reading unit 21, a TDMS information reading unit 22, a control memory 23, a defect entry adding unit 24, a TDFL conversion unit 25, a TDMS information writing unit 26, a verifying unit 27, a position  
5 information generating unit 28, and a DMA information writing unit 29.

[0048]

<DMA information reading unit 21>

The DMA information reading unit 21 determines normal DMAs  
10 among the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA, and reads the contents of the normal DMAs onto the control memory 23. It should be noted here that the DMA information reading unit 21 judges that it is "after the finalize process" if the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA can be reproduced normally, and judges that it is "before the finalize process"  
15 if data cannot normally be read from the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA since no data has been recorded therein.

[0049]

<TDMS information reading unit 22>

The TDMS information reading unit 22, if the optical disc  
20 1 loaded in the recording/reading apparatus 100 is "not-finalized", searches for a last cluster in which data is recorded, and searches for the latest TDMS (TDMS#j-1). The TDMS information reading unit 22 then extracts the TDMS from the last cluster in which data is recorded, and acquires the number of TDFLs, which is  
25 represented as "n", and positions of the TDFLs. Using the acquired positions, the TDMS information reading unit 22 reads TDFLs from the latest TDMS onto the control memory 23, and holds them as a plurality of defect entries that constitute a DFL. The plurality of defect entries are used in the read process.

[0050]

<Control Memory 23>

The control memory 23 is a memory for storing data for work, such as defect entries.

5 <Defect Entry Adding Unit 24>

The defect entry adding unit 24, if a defective area is detected in the  $j^{\text{th}}$  record process, adds a defect entry concerning the detected defective area to a plurality of defect entries having been read onto the control memory 23. More specifically, if, 10 for example, a defective cluster is newly detected in the user data area 6, the defect entry adding unit 24 adds a defect entry corresponding to the newly detected defective cluster to a plurality of defect entries stored in the control memory 23. Further, according to the defective cluster position information 15 included in the defect entry, the defect entry adding unit 24 sorts defect entries, and increases the number of defect entries by "1".

<TDFL Conversion Unit 25>

The TDFL conversion unit 25 arranges (i) defect entries 20 concerning defective areas detected in the latest record process, and (ii) defect entries concerning defective areas detected in the other previous record processes, in the ascending order of the addresses of the defective areas indicated by these defect entries, and converts the arranged defect entries into "TDFLs". 25 This achieves an update of TDMSs. During this conversion, the TDFL conversion unit 25 calculates the total size of the plurality of defect entries stored in the control memory 23, and judges whether or not a result of adding the total size to the fixed size of the TDDS exceeds the size of one cluster. If the addition



result is smaller than the cluster size, the TDFL conversion unit 25 converts the plurality of defect entries into one TDFL. When a TDMS is composed of one cluster, the size of the cluster should be a result of adding up the size of TDFL#1 and the fixed size of the TDDS. Here, since the size of the TDDS is identical with the size of one sector, the TDFL is identical with 31 sectors in size at the largest. If the addition result is no smaller than the cluster size, the TDFL conversion unit 25 converts the plurality of defect entries into a plurality of TDFLs.

10 [0051]

<TDMS Information Writing Unit 26>

The TDMS information writing unit 26 writes the TDFLs obtained as a result of the conversion by the TDFL conversion unit 25, into the latest TDMS (TDMS#j) in the ascending order of the addresses of the defective areas indicated by the TDFLs. After writing the TDFLs, the TDMS information writing unit 26 also writes the TDDS into TDMS#j. Figs. 11A to 11D show reading of TDMS information by the TDMS information reading unit 22 and writing of TDMS information by the TDMS information writing unit 26. Fig. 11A shows the state before the reading/writing of TDMS information is performed. In Fig. 11A, TDFLs and TDDS have been written in TDMS#1 to TDMS#j-1 among the N number of TDMSs, and only TDMS#j-1 is valid. In this case, the TDMS information reading unit 22, in the j<sup>th</sup> recording, reads TDFL#1 to TDFL#3 and the TDDS from TDMS#j-1 (Fig. 11B). It is supposed here that a defective area is detected in the j<sup>th</sup> recording. TDFL#4 concerning this defective area is combined with TDFL#1 to TDFL#3 written in TDMS#j-1 (Fig. 11C). The TDMS information writing unit 26 writes TDFL#1 to TDFL#4 and the TDDS into TDMS#j (Fig. 11D). TDMS#j at this

stage includes the TDFLs that show all the defective areas that were detected in the first to  $j^{\text{th}}$  recordings.

[0052]

#### <Verifying Unit 27>

5           The verifying unit 27 is a component embedded in the TDMS information reading unit 22, and verifies the TDFLs written by the TDMS information writing unit 26. In the present document, the term "verify" indicates an operation to confirm whether or not recording data into a cluster has been completed normally.

10   The verifying is achieved by performing an error correction onto the recorded data, or by reading the recorded data and judging whether or not the read data matches the preliminarily recorded data. If it is judged that the data is not recorded normally due to a defect or the like, the TDMS information writing unit

15   26 retries writing of TDFLs.

[0053]

#### <Position Information Generating Unit 28>

          The position information generating unit 28, when the TDMS information writing unit 26 tries to write TDFLs into TDMS#j,

20   creates, in the TDDS of the control memory 23, the TDFL position information that indicates write destinations of the TDFLs. The position information generating unit 28 creates the TDFL position information when writing of TDFLs is retried, as well as in the initial writing of the TDFLs. After a plurality of pieces of

25   TDFL position information are generated in the TDDS on the control memory 23, only pieces of TDFL position information concerning the normally recorded TDFLs are extracted from the plurality of pieces of TDFL position information, and the extracted pieces of TDFL position information are arranged in the ascending order

of the addresses of the defective areas indicated by the corresponding TDFLs. With this construction, even if normal TDFLs are discretely arranged in TDMS#j due to a retried writing of TDFLs, only pieces of TDFL position information concerning the normally recorded TDFLs are extracted, and they are arranged in the ascending order of the addresses of the defective areas indicated by the corresponding TDFLs.

[0054]

As described above, even if normal TDFLs are arranged in TDMS#j discretely, the position information generating unit 28 rearranges the plurality of pieces of TDFL position information so that pieces of TDFL position information corresponding to the normal TDFLs arranged in TDMS#j discretely can be read out in a predetermined order. When the TDMS information writing unit 26 writes the last TDFL, the TDDS is written into the TDMS together with the last TDFL.

[0055]

#### <DMA Information Writing Unit 29>

The DMA information writing unit 29 records data into the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA during the finalize process. Figs. 12A to 12C show writing of information by the DMA information writing unit 29. Fig. 12A shows the state before the writing is performed. In Fig. 12A, TDMS#1 to TDMS#4 and TDDS have been written in TDMS#j among the N number of TDMSs, and only the latest TDMS#j is valid. In this case, the contents of TDMS#j are read by the DMA information reading unit 21 onto the control memory 23 (Fig. 12B), defect entries written in TDFL#1 to TDFL#4 in TDMS#j are converted into DFLs by the DMA information writing unit 29, and the data after the conversion is written into the 1<sup>st</sup> DMA (Fig. 12C). Similarly,

the DFLs converted from TDFL#1 to TDFL#4 are written into the 2<sup>nd</sup> DMA, 3<sup>rd</sup> DMA, and 4<sup>th</sup> DMA.

[0056]

Up to now, the internal structure of the defect management information processing unit 13 has been described. From now on, implementing the defect management information processing unit 13 from software will be described. The defect management information processing unit 13 can be implemented in the recording/reading apparatus 100 by generating a program based on the flowcharts shown in Figs. 13 to 15, and causing the CPU to run the program.

Fig. 13 is a flowchart showing the procedure of the DFL read process.

[0057]

In step S1, it is checked whether or not the optical disc 1 has been loaded. If it is confirmed that the optical disc 1 has been loaded, in step S2, the DMA information reading unit 21 judges whether or not the optical disc 1 has been finalized. If the optical disc 1 has been finalized, the optical disc 1 is subjected into a process that is the same as a process for a read-only disc.

If the optical disc 1 has not been finalized, the TDMS information reading unit 22 searches a plurality of TDMSs for the last cluster among those in which data is recorded, and searches for the latest TDMS, TDMS#j (step S3). The TDMS information reading unit 22 then extracts the TDDS from the last cluster among those in which data is recorded, and acquires the number of TDFLs, which is represented as "n", and positions of the TDFLs (step S5). After acquiring the positions, the TDMS information reading

unit 22 executes the loop composed of steps S6 to S9. In this loop, it initializes the variable k (step S6), reads TDFL#k from TDMS based on the TDFL#k position information among a plurality of pieces of TDFL position information in the TDDS (step S7).

5 In the present flowchart, the variable k is used as a control variable. The variable k is incremented by one each time a loop composed of steps S7 to S9 is performed (step S9). The control goes out of the loop when the variable k becomes n (step S8).  
[0058]

10 After the TDFLs are read as described above, defect entries constituting the TDFLs are held by the control memory 23 as defect entries constituting the DFLs (step S10). Then, the read or write process for reading or writing data from/to the optical disc 1 is performed in accordance with a command issued from the high-order  
15 control apparatus 200 (step S30).

Fig. 14 is a flowchart showing the procedure of the read/write process to be performed in accordance with a command issued from the high-order control apparatus 200.

[0059]

20 In Fig. 14, the steps S11 and S12 constitute a command wait loop. In step S11, the instruction processing unit 12 judges whether or not the command issued from the high-order control apparatus 200 is the read command. In step S12, the instruction processing unit 12 judges whether or not the command issued from  
25 the high-order control apparatus 200 is the write command.

If the command issued from the high-order control apparatus 200 is the read command, the steps S13 to S15 are performed. In step S13, the instruction processing unit 12 judges whether or not the read destination of the command is a defective cluster

indicated by any defect entry in the DFL. If the read destination is not a defective cluster, user data is read from the read destination (step S14). If the read destination is a defective cluster, the defect management information processing unit 13  
5 reads user data from a substitute cluster for the defective cluster (step S15), and returns to the loop composed of the steps S11 to S12.

[0060]

If the command issued from the high-order control apparatus  
10 200 is the write command, user data is written into the write destination cluster (step S18). After this, the control moves to a loop composed of the steps S19 to S23.

In this loop, the following steps (steps S19-S23) are repeatedly performed until it is judged as "No" in step S20. First,  
15 the written user data is verified (step S19). If the verification result indicates a detection of a defective cluster (Yes in step S20), the defect management information processing unit 13 assigns a substitute cluster to the defective cluster (step S21), and generates a defect entry that shows correspondence between the  
20 defective cluster and the substitute cluster (step S22), and the user data is written into the substitute cluster as a retry (step S23).

[0061]

If the verification of the written user data does not detect  
25 any defective cluster, a process composed of the steps S24 to S26 is performed as follows. First, it is judged whether or not a defect entry has been generated newly (step S24). If a defect entry has been generated newly, the DFL is updated and the new defect entry is added (step S25). The DFL after the update is

written into the latest TDMS (step S26). The control then returns to the loop composed of the steps S11 to S12.

[0062]

Fig. 15 is a flowchart showing the procedure of the write-to-TDMS process.

In step S31, the TDFL conversion unit 25 converts a plurality of defect entries included in the DFLs into one or more TDFLs. In step S32, the start position of a cluster, which has a larger serial number than the last data-recorded cluster in the TDMA of the control memory 23, is obtained, where the cluster of the obtained start position is the first cluster of the not-recorded area. The first cluster of the not-recorded area is regarded as the current cluster. The recording starts with the current cluster.

[0063]

After this, a loop composed steps S33 to S40 is performed.

In this loop, not-recorded TDFLs are fetched one by one in the ascending order from the control memory 23, and the TDMS information writing unit 26 writes the fetched TDFL into the current cluster as TDFLi (step S33). Then, the next cluster is specified as the current cluster (step S34), and the variable I is incremented (step S35). The loop is repeated until TDFLi becomes TDFLn-1, which is the TDFL immediately before the last TDFL (Yes in step S36).

[0064]

In this loop, each time one TDFL is written, the verifying unit 27 verifies the writing (step S37). If the verification reveals that a cluster in which the TDFL is written is defective, the variable i is not incremented, but the retry flag is turned

ON (step S38). In step S39 positioned before the step S33, it is judged whether the retry flag is ON or OFF. If it is judged that the retry flag is ON, the TDMS information writing unit 26 writes the TDFLi that failed to be recorded, into the current  
5 cluster (step S40).

[0065]

When TDFL#1 to TDFL#3 (the TDFL before the last TDFL) have been written in the TDMS after the above-stated steps are repeated, steps S41 to S45 are performed.

10 The steps S41 to S45 constitute a process in which the last TDFL and the TDDS are written into the TDMS. In step S41, the position information generating unit 28 generates, in the TDDS, TDFL position information that indicates the positions of the already-written TDFLs and the position to write the last TDFL  
15 (step S41). Then a loop composed of steps S42 to S45 is performed.

[0066]

The loop is performed as follows. That is to say, the TDMS information writing unit 26 writes the generated TDDS and the last TDFL into the current cluster (step S42), and the verifying  
20 unit 27 verifies the cluster in which the data was written (step S43). If the cluster is not defective, the process ends. If the cluster is defective, the next cluster is specified as the current cluster (step S44). The position information generating unit 28 then newly generates a TDDS by rewriting the position of the  
25 last TDFL in the TDDS (step S45). The steps S42 and S43 are then performed again.

[0067]

The rewriting of the TDFL position information by the position information generating unit 28 in step S45 is achieved



by writing the address of the current cluster into the TDFL position information of the last TDFL.

Fig. 16 shows how the recording/reading apparatus 100 of the first embodiment writes the TDFLs. The first row of Fig. 16 shows a plurality of clusters that constitute a TDMS. The second to sixth rows show four TDFLs (TDFL#1, TDFL#2, TDFL#3, and TDFL #4) and a TDDS written in these clusters. It is presumed that cluster#2 shown in the first row is a defective cluster. [0068]

The second row indicates the state in which TDFL#1 has been written in cluster#1 among the clusters constituting the TDMS. The third row indicates that writing TDFL#2 into cluster#2 was tried, but failed since cluster#2 is a defective cluster. This causes a retry to be made to record TDFL#2. The fourth row indicates that TDFL#2 is written in cluster#3 that is adjacent to the defective cluster#2, in a recording retry. The fifth row indicates that TDFL#3 has been written in cluster#4. The sixth row indicates that the last TDFL (TDFL#4) and the TDDS have been written in cluster#5.

[0069]

Although TDFL#1 and TDFL#2 to TDFL#4 are arranged discretely with a defect entry in between, the TDDS written together with the last TDFL indicates the positions of the clusters (addresses C1, C3, C4, C5) in which TDFL#1, TDFL#2, TDFL#3, and TDFL#4 are written. With such a construction, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the recording/reading apparatus 100, only by accessing the clusters in the TDMS in accordance with the positions indicated by the TDDS.

[0070]

As described above, according to the present embodiment in which the order of reading TDFLs is indicated by TDFL position information, it is not necessary to arrange a plurality of TDFLs sequentially even if spare areas have a defective cluster. The TDFLs are arranged in discrete sequential areas including the defective cluster. That is to say, the present embodiment enables the TDFLs to be arranged randomly.

With this construction in which TDFLs can be arranged randomly, when only one defective cluster is detected while a plurality of TDFLs are written into a sequence of clusters in the TDMS, it is not necessary to retry writing all the plurality of TDFLs from the start, and only the TDFL failed to be written into the defective cluster is written into another cluster as a retry. Since this reduces the number of TDFLs to be written as a retry, it is possible to prevent the TDMS from being consumed rapidly.

#### <Second Embodiment>

The second embodiment proposes an arrangement of TDFLs in the TDMS to increase the speed of writing into the TDMS.

[0071]

Fig. 17 shows an example of a discrete arrangement of TDFLs in the second embodiment. The first row of Fig. 17 shows a plurality of clusters that constitute a TDMS. The second row shows four TDFLs (TDFL#1, TDFL#2, TDFL#3, and TDFL#4) and a TDDS written in these clusters.

In the second row, the TDFLs and TDDS are arranged in the order of TDFL#1, TDFL#3, TDFL#4, TDDS, TDFL#2, TDFL#4, and TDDS.

There are two TDFL#4s and two TDDs. Among these TDFL#4s and TDDs, the TDFL#4 and TDD recorded later are valid, and the TDFL#4 and TDD having been recorded earlier are invalid. This is because only a TDD that is immediately before the not-recorded area is made valid.

[0072]

It would be understood that in TDMS#j shown in Fig. 17, TDFL#1 to TDFL#4 are arranged in the following order:

- i) TDFLs written in the initial writing (TDFL#1, TDFL#3);
- ii) TDFLs written as a retry (TDFL#2); and
- iii) the last TDFL (TDFL#4).

[0073]

In each TDD, a plurality of pieces of TDFL position information are arranged in the ascending order of the addresses of the defective areas indicated by the corresponding TDFLs.

In Fig. 17, cluster#2 is a defective cluster. Basically, TDFL#2, which failed to be written into the defective cluster, should be written in a normal cluster to follow TDFL#1. However, in this example, TDFL#2 is written, as a retry, in a cluster that is followed by a cluster in which TDFL#4 and TDD are written. Even if, as in this case, two TDFL#4s are arranged, and TDFL#2 is arranged in no particular order, the TDD indicates the positions of the clusters (addresses C1, C5, C3, and C6) in which valid TDFL#1 to TDFL#4 are written in the order of TDFL#1, TDFL#2, TDFL#3, and TDFL#4. With such a construction, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the recording/reading apparatus 100, only by accessing the clusters in the TDMS in accordance with the positions indicated by the TDD. As such, the improvement

in the second embodiment regarding the optical disc is to allow not only a discrete arrangement of TDFLs, but also a redundant arrangement of TDFLs.

[0074]

5           Upto now, the improvement in the second embodiment regarding the optical disc has been described. The following describes the recording/reading apparatus 100 in the second embodiment. The improvement in the recording/reading apparatus 100 of the second embodiment is achieved by the TDMS information writing  
10   unit 26, verifying unit 27, and position information generating unit 28.

          The TDMS information writing unit 26 in the second embodiment writes all the TDFLs and TDDSs at once into the latest TDMS in the TDMA. If a cluster in which a TDFL has been written is a  
15   defective cluster, the TDMS information writing unit 26 writes the TDFL written in the defective cluster, the last TDFL, and the TDDS as a retry.

[0075]

          The verifying unit 27 in the second embodiment verifies  
20   the TDFLs written by the TDMS information writing unit 26 (including the TDFLs written as a retry) to judge whether or not the TDFLs have been written correctly.

          The position information generating unit 28 in the second embodiment, when the TDMS information writing unit 26 writes TDFLs  
25   (including a retry), generates TDFL position information to be stored in the TDDS.

[0076]

          The improvement achieved by the TDMS information writing unit 26, verifying unit 27, and position information generating

unit 28 is disclosed in the flowchart shown in Fig. 18.

Fig. 18 is a flowchart showing the procedure of the write-to-TDMS process in the second embodiment. In this flowchart, after steps S31 and S32 are performed as in the flowchart of Fig.

5 15, steps S51 to S52 are performed.

[0077]

In steps S51 to S52, the position information generating unit 28 generates in the TDDS the TDFL position information which indicates the positions of four clusters (starting with the current  
10 cluster) predicted to store a plurality of TDFLs, (step S51), and TDFL#1 to TDFL#4 and TDDS are written into the current cluster and after (step S52). It should be noted here that by recording a sequence of TDFLs into a sequence of clusters, it is possible to reduce the wait time that occurs when the recording start  
15 position is searched for, and to reduce the time required for updating the TDMS.

[0078]

In step S51, the position information generating unit 28 generates the TDFL position information based on the prediction  
20 that four TDFLs are written in the current cluster and the succeeding clusters. Here, suppose that there are the k number of TDFLs, then any of the k number of TDFLs might be written as a retry, and the following relationship is used to detect the target of the retry:

25 TDFL position information of the  $i^{\text{th}}$  TDFL ( $i \leq k$ )  
 $\leftarrow$  current cluster + (i-1) \* clusters

In step S53, the verifying unit 27 verifies the results of the above-described writing of all TDFLs. If it is judged that all the TDFLs have been written correctly (Success in step

S53), the process of this flowchart ends.

[0079]

If a defective cluster is detected (Failure in step S53), the position information generating unit 28 generates in the TDDS  
5 the TDFL position information that shows positions of already-written TDFLs, positions to write TDFLs that failed to be written, and a position to write the last TDFL (step S54), then a loop composed of steps S55 to S58 is performed.

In this loop composed of steps S55 to S58, the TDMS information  
10 writing unit 26 makes a retry. In the example shown in Fig. 17, the targets of the retry are not only TDFL#2 failed to be written, but a new TDDS that indicates a new position of TDFL#2, and TDFL#4 to be written in the same cluster as the TDDS.

[0080]

15 The steps S55 to S58 are performed as follows. The TDMS information writing unit 26 writes the TDFL that failed to be written, the last TDFL, and the TDDS, into the current cluster and the succeeding clusters (step S55). The verifying unit 27 verifies the clusters into which the data has been written (step  
20 S56). If the clusters into which the data has been written are not defective, the process ends. If the current cluster into which the data has been written is a defective cluster, the next cluster is specified as the current cluster (step S57). The position information generating unit 28 newly generates a TDDS  
25 by rewriting the position the TDFL that failed to be written, and the position of the last TDFL in the TDDS (step S58). The steps S55 and S56 are performed again.

[0081]

In steps S54 and S58, the position information generating

unit 28 generates or rewrites the TDFL position information as follows. In a retry of writing TDFLs, the targets of the retry are arranged in the TDMS in the following order: (1) one or more TDFLs; and (2) the last TDFL. As a result, the position information  
5 generating unit 28 rewrites the TDFL position information in the TDDS on presumption that the one or more TDFLs as the targets of the retry are written into the current cluster and the succeeding clusters. Here, suppose that the number of the one or more TDFLs as the targets of the retry is "k", the following relationships  
10 are used to detect the targets of the retry:

TDFL position information of the  $i^{\text{th}}$  TDFL ( $i \leq k$ )

$\leftarrow$  current cluster + (i-1) \* clusters; and

TDFL position information of the last TDFL

$\leftarrow$  current cluster + (k-1) \* clusters.

15 Fig. 19 shows the process in which TDFLs are written by the recording/reading apparatus 100 of the second embodiment. The first row of Fig. 19 shows a plurality of clusters that constitute TDMS#j. The second and third rows show the process in which four TDFLs (TDFL#1, TDFL#2, TDFL#3 and TDFL#4) and a  
20 TDDS are written in these clusters. In the first row, cluster#2 is a defective cluster. The second row indicates the state in which TDFL#1-TDFL#4 and TDDS have been written in cluster#1-cluster#4. Here, the TDDS indicates the positions of the clusters (addresses C1, C2, C3, and C4) in which TDFL#1, TDFL#2, TDFL#3, and TDFL#4 have been written. A retry is required since,  
25 as described above, cluster#2 is a defective cluster. Here, the targets of the retry are not only TDFL#2 failed to be written, but a new TDDS that indicates a new position of TDFL#2, and TDFL#4 to be written in the same cluster as the TDDS. The third row

indicates the state in which TDFL#2, TDFL#4, and TDDS have been written into cluster#5 and the succeeding clusters, as a result of the retry. The TDDS written together with the last TDFL indicates the positions of the clusters (addresses C1, C5, C3, and C6) in which TDFL#1, TDFL#2, TDFL#3, and TDFL#4 have been written. With such a construction, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the recording/reading apparatus 100, only by accessing the clusters in the TDMS in accordance with the positions indicated by the TDDS.

[0082]

As described above, according to the second embodiment in which TDFLs that failed to be recorded due to a defect or the like, the last TDFL, and the TDDS are written as a retry, it is possible to reduce the time required to update the TDMS, improving the facility of use. Further, it is possible to reduce the risk of failure in updating TDMS which may occur, for example, if a power is turned off abruptly during the update.

Also, a defect such as a flaw in TDMA often involves a plurality of defective clusters that align in the radius direction. In the optical disc, one inner circumference approximately corresponds to two clusters. Accordingly, if the TDMS is composed of four clusters, and if always the same contents are tried to be written into the TDMS as a retry, there is a possibility that the update of the TDMS may repeatedly fail. However, according to the procedure of the second embodiment, if defective clusters align in the radius direction, TDFLs to be written into the defective clusters change for each retry. This increases the probability that a retry is performed successfully.



[0083]

### Third Embodiment

The third embodiment relates to an improvement that achieves a speed of writing that is approximately in the middle of those of the first and second embodiments.

Fig. 20 shows an example of a discrete arrangement of TDFLs in the third embodiment. The first row of Fig. 20 shows a plurality of clusters that constitute a TDMS. The second row shows four TDFLs (TDFL#1, TDFL#2, TDFL#3 and TDFL#4) and a TDDS written in these clusters.

[0084]

In the second row, the TDFLs and TDDS are arranged in the order of TDFL#1, TDFL#3, TDFL#2, TDFL#4, and TDDS. In Fig. 20, cluster#2 is a defective cluster, and TDFL#2 written in the defective cluster is written in a cluster before TDFL#4 that is the last TDFL. It would be understood that TDFL#2 was written as a retry in a cluster before TDFL#4, the last TDFL. Even if, as in this case, TDFL#2 is arranged in no particular order, the TDDS indicates the positions of the clusters (addresses C1, C4, C3, and C5) in which TDFL#1, TDFL#2, TDFL#3, and TDFL#4 have been written in the order of TDFL#1 to TDFL#4. With such a construction, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the recording/reading apparatus 100, only by accessing the clusters in the TDMS in accordance with the positions indicated by the TDDS. As such, the improvement in the third embodiment regarding the optical disc is to allow a discrete arrangement of TDFLs.

[0085]

As described above, in the optical disc 1 in the third

embodiment, a plurality of pieces of TDFL position information are arranged in the TDDS in the ascending order of the defect entries, and only the start position information of the TDFLs that failed to be recorded due to a defect or the like are arranged to follow the plurality of pieces of TDFL position information, enabling the TDMS, which has a finite size, to be used efficiently.

Up to now, the improvement in the third embodiment regarding the optical disc has been described. The following describes the recording/reading apparatus 100 in the third embodiment. The improvement in the recording/reading apparatus 100 of the third embodiment is achieved by the TDMS information writing unit 26, verifying unit 27, and position information generating unit 28.

[0086]

The TDMS information writing unit 26 in the third embodiment writes all the TDFLs, except for the last TDFL, at once into the latest TDMS in the TDMA. If a cluster in which a TDFL has been written is a defective cluster, the TDMS information writing unit 26 writes the TDFL written in the defective cluster, as a retry, then writes the last TDFL and the TDDS into the TDMS.

The verifying unit 27 in the third embodiment verifies the TDFLs written by the TDMS information writing unit 26 (including the TDFLs written as a retry) to judge whether or not the TDFLs have been written correctly.

[0087]

The position information generating unit 28 in the third embodiment, when the TDMS information writing unit 26 writes TDFLs (including a retry), generates TDFL position information to be stored in the TDDS.

The improvement achieved by the TDMS information writing

unit 26, verifying unit 27, and position information generating unit 28 is disclosed in the flowchart shown in Fig. 21. The following describes the process performed by the TDMS information writing unit 26, verifying unit 27, and position information  
5 generating unit 28 of the third embodiment, with reference to the flowchart.

[0088]

Fig. 21 is a flowchart showing the procedure of the write-to-TDMS process in the third embodiment. In this flowchart,  
10 after steps S31 and S32 are performed as in the flowchart of Fig. 15, steps S63 to S64 are performed.

In steps S63 to S64, the position information generating unit 28 generates in the TDDS the TDFL position information which indicates the positions of four clusters (starting with the current  
15 cluster) predicted to store a plurality of TDFLs, (step S63), and TDFL#1 to TDFL#n-1 are written into the current cluster and after (step S64).

[0089]

In step S65, the verifying unit 27 verifies the results  
20 of the above-described writing of all TDFLs. If it is judged that all the TDFLs have been written correctly (Success in step S65), the process composed of steps S66 to S70 is skipped.

The steps S66 to S70 are performed if a defective cluster is detected. The position information generating unit 28  
25 generates in the TDDS the TDFL position information that shows positions of already-written TDFLs, and positions to write TDFLs that failed to be written (step S66), then a loop composed of steps S67 to S70 is performed.

[0090]

writing

The steps S67 to S70 are performed as follows. The TDMS information writing unit 26 writes the TDFLs that failed to be written into the current cluster and the succeeding clusters (step S67). The next cluster is specified as the current cluster (step S68). The verifying unit 27 verifies the clusters into which the data has been written (step S69). If the clusters into which the data has been written are not defective, the control goes out of the loop to step S71. If the current cluster into which the data has been written is a defective cluster, the position information generating unit 28 rewrites the position information of the TDFLs that failed to be written, in the TDDS (step S70). The control then moves to step S67 again.

[0091]

In step S70, the position information generating unit 28 rewrites the TDFL position information as follows. In a retry of writing TDFLs, the position information generating unit 28 rewrites the TDFL position information in the TDDS on presumption that one or more TDFLs as the targets of the retry are written into the current cluster and the succeeding clusters. Here, suppose that the number of the one or more TDFLs as the targets of the retry is "k", the following relationships are used to detect the targets of the retry:

TDFL position information of the  $i^{\text{th}}$  TDFL ( $i \leq k$ )

$\leftarrow$  current cluster + (i-1) \* clusters.

The steps S71 to S75 are performed as follows. The position information generating unit 28 generates in the TDDS the TDFL position information that shows positions of already-written TDFLs, and the position to write the last TDFL at (step S71), then a loop composed of steps S72 to S75 is performed.

[0092]

The loop is performed as follows. That is to say, the TDMS information writing unit 26 writes the generated TDDS and the last TDFL into the current cluster (step S72), and the verifying unit 27 verifies the cluster in which the data was written (step S73). If the cluster is not defective, the process ends. If the cluster is defective, the next cluster is specified as the current cluster (step S74). The position information generating unit 28 then newly generates a TDDS by rewriting the position of the last TDFL in the TDDS (step S75). The steps S72 and S73 are then performed again.

[0093]

The rewriting of the TDFL position information by the position information generating unit 28 in step S75 is achieved by writing the address of the current cluster into the TDFL position information of the last TDFL.

Fig. 22 shows how the recording/reading apparatus 100 of the third embodiment writes the TDFLs. The first row of Fig. 22 shows a plurality of clusters that constitute TDMS#j. The second to fourth rows show four TDFLs (TDFL#1, TDFL#2, TDFL#3, and TDFL #4) and a TDDS written in these clusters. It is presumed that cluster#2 shown in the first row is a defective cluster, and that the second row shows the state in which TDFL#1 to TDFL#3 have been written in cluster#1 to cluster#3.

25 [0094]

As described above, cluster#2 is a defective cluster. Therefore, a retry is required. In this retry, only TDFL#2, which failed to be written in the initial writing, is written. However, at the same time, TDFL#4 being the last TDFL and the TDDS are

written all at once. The second row indicates the state in which TDFL#2 has been written as a retry, and the fourth row indicates the state in which TDFL#4 and TDDS have been written. The TDDS written together with the last TDFL indicates the positions of  
5 the clusters (addresses C1, C4, C3, C5) in which TDFL#1, TDFL#2, TDFL#3, and TDFL#4 are written. With such a construction, it is possible, during reproduction, to read out the TDFLs sequentially in a predetermined order onto the memory in the recording/reading apparatus 100, only by accessing the clusters  
10 in the TDMS in accordance with the positions indicated by the TDDS.

[0095]

As described above, according to the third embodiment in which TDFLs that failed to be recorded due to a defect or the  
15 like, the last TDFL, and the TDDS are written as a retry, it is possible to reduce the time required to update the TDMS, improving the facility of use. Further, it is possible to reduce the risk of failure in updating TDMS which may occur, for example, if a power is turned off abruptly during the update.

20 Also, a defect such as a flaw in TDMA often involves a plurality of defective clusters that align in the radius direction. In the optical disc, one inner circumference approximately corresponds to two clusters. Accordingly, if the TDMS is composed of four clusters, and if always the same contents are tried to  
25 be written into the TDMS as a retry, there is a possibility that the update of the TDMS may repeatedly fail. However, according to the procedure of the third embodiment, if defective clusters align in the radius direction, TDFLs to be written into the defective clusters change for each retry. This increases the

probability that a retry is performed successfully.

#### <Supplementary Notes>

Up to now, the best mode for carrying out the invention, as far as known to the applicant at this time of filing the present application, has been described. However, further improvements or modifications can be made on the present invention in terms of the following technical topics. It should be noted here that whether or not to make such improvements or modifications is optional, and depends on the implementer of the invention.

#### 10 <Construction of TDMS>

In the above-described embodiments, the TDMS is composed of TDFLs and TDDs. However, other information may be added to the TDMS. In that case, a TDMS is equivalent, in size, to the total of the TDFLs, TDDs, and the additional information constituting the TDMS.

[0096]

#### <TDDs>

The TDDs may not necessarily be the same as the DDS in size. It will naturally be understood that when the total of TDFLs, TDDs, and other meaningful information does not reach a cluster in size, the remaining portion of the cluster is filled with such data that is not meaningful, such as "0".

#### <Application to Other Lists>

In the first embodiment, the TDFL is explained. However, it will naturally be understood that the TDFL, as a target of the present invention, may be replaced with another list that, for example, is composed of replacement source entries and replacement destination entries, where data A to overwrite data B in a recorded area is recorded in another area, and an entry

is used to virtually overwrite.

#### <Updating TDMS>

In the first embodiment, if a new addition of a defect entry affects only TDFLs that constitute part of a TDMS, only the affected  
5 part of the TDMS, not the whole TDMS, may be updated.

#### <Finalizing Indication Flag>

The DMA information reading unit 21 has a function to judge the 1<sup>st</sup> DMA to the 4<sup>th</sup> DMA whether or not the disc has been finalized. However, if a finalizing indication flag, which indicates whether  
10 or not the disc has been finalized, is provided at a predetermined position in the TDMS, the TDMS information reading unit 22 can judge, using the flag, whether or not the disc has been finalized. The finalizing indication flag does not necessarily be included in the TDMS, but may be provided at any predetermined position  
15 in the medium.

#### <Conversion from TDFL to DFL>

In the above describe embodiments, TDFLs are read out from a plurality of clusters on a one-cluster-to-one-cluster basis in an order indicated by the TDFL position information, and are  
20 stored in the storage buffer 16. However, not limited to this, after TDFLs are read out from all the clusters ranging from the one at the position indicated by the TDFL#1 position information to the last cluster in the recorded area, only valid TDFLs corresponding to the positions indicated by the TDFL position  
25 information are extracted, and the extracted TDFLs may be arranged in an order indicated by the TDFL position information.

#### <Reading TDFLs>

In the above describe embodiments, TDFLs are read out from a plurality of clusters on a one-cluster-to-one-cluster basis



in an order indicated by the TDFL position information, and are stored in the storage buffer 16. However, not limited to this, it may be judged from the continuity of pieces of TDFL position information whether or not any defective clusters or invalid clusters are included in the clusters in which the TDFLs corresponding to the positions indicated by the TDFL position information are recorded, and if TDFLs are recorded in valid clusters in succession, the valid TDFLs may be read out in succession from the clusters. This construction reduces the time required to obtain defect entries, compared with the case where TDFLs are read out from a plurality of clusters on a one-cluster-to-one-cluster basis.

[0097]

#### <Conversion from TDFL to DFL>

In the storage buffer 16, a combination of all the read-out TDFLs may be stored as DFL, or the read-out TDFLs may be stored as they are.

#### <Updating TDMS>

The update of the TDMS performed by the TDFL conversion unit 25 in the first embodiment is merely an example. The update may be performed differently in so far as information regarding defects is reflected. For example, sorting the defect entries may not necessarily be performed.

#### <Construction of TDMS>

In the above-described embodiments, the TDMS is composed of TDFLs and TDDs. However, other information may be added to the TDMS. In that case, a TDMS is equivalent, in size, to the total of the TDFLs, TDDs, and the additional information constituting the TDMS.

### <Timing of Verifying>

In the second and third embodiments, after TDFLs including those that failed to be written in the initial writing due to a defective cluster are written and the last TDFL and the TDDS are written, the verification is performed. However, not limited to this, for example, a process, in which only TDFLs that failed to be written due to a defective cluster are written and a verification is performed, may be repeated until all the TDFLs except for the last TDFL are successfully written, and then the last TDFL and the TDDS may be written and a verification may be performed.

### <Occurrence of Error during Recording>

In the second and third embodiments, if it is judged during the verification that a recording failure occurred, the recording is performed as a retry. However, not limited to this, for example, a retry may be performed as an error occurs during writing of data. In this case, the following process may be repeated to obtain the same advantageous effects: if an error occurs during writing of data, clusters in which data has been written in succession are verified, TDFLs including those that failed to be written, and the last TDFL and the TDDS are written, and a verification is performed.

### <Updating TDMS>

In the third embodiment, if a new addition of a defect entry affects only TDFLs that constitute part of a TDMS, only the affected part of the TDMS, not the whole TDMS, may be updated.

### <Rewriting TDDS>

In the second embodiment, before a retry is made, each piece of TDFL position information corresponding to each TDFL is obtained

and the TDFL position information in the TDDS is rewritten with the obtained TDFL position information. However, the TDFL position information in the TDDS may not necessarily be rewritten before a retry is made. For example, each piece of TDFL position information corresponding to each valid TDFL may be obtained when the last TDFL and the TDDS are written, and the TDFL position information in the TDDS may be rewritten with the obtained TDFL position information.

#### <Timing of Verifying>

10 In the third embodiment, a process, in which the verification is performed after TDFLs including those that failed to be written are written normally, is repeated until all the TDFLs except for the last TDFL and the TDDS are written normally, and then after the last TDFL and the TDDS are written, the verification is performed. However, TDFLs that failed to be written and the TDDS may be written into clusters that align in sequence, and then the verification may be performed.

[0098]

#### <Timing of Verifying>

20 In the third embodiment, a process, in which the verification is performed after TDFLs including those that failed to be written are written normally, is repeated until all the TDFLs except for the last TDFL and the TDDS are written normally, and then after the last TDFL and the TDDS are written, the verification is performed. However, process A, in which a TDFL that failed to be written is written and the verification is performed, may be repeated until all the TDFLs except for the last TDFL are written normally, and then process B, in which after the last TDFL and the TDDS are written, the verification is performed, may be

performed. The sequence of processes A and B may be repeated.

#### <Order of TDFLs>

Even in the case where a defective cluster is included in the clusters, the order of valid TDFL#1 to TDFL#n may be an ascending order. For example, the cluster in which TDFL#n was written is a defective cluster, the TDFLs are written in the order of TDFL#1, TDFL#2, TDFL#3, . . . [defective cluster], TDFL#n, which is an ascending order.

#### <Realization of Control Procedure>

Both the control procedures explained in the above-described embodiments using the flowcharts and the control procedures of the functional components explained in the above-described embodiments satisfy the requirements for the "program invention" since the above-mentioned control procedures are realized concretely using the hardware resources and are the creation of a technical idea utilizing natural laws.

[0099]

#### ■ Production of Program of Present Invention

The program of the present invention can be produced as follows. First, the software developer writes, using a programming language, a source program that achieves each flowchart and functional component. In this writing, the software developer uses the class structure, variables, array variables, calls to external functions, and so on, which conform to the sentence structure of the programming language he/she uses.

[0100]

The written source program is sent to the compiler as files. The compiler translates the source program and generates an object program.

The translation performed by the compiler includes processes such as the sentence structure analysis, optimization, resource allocation, and code generation. In the sentence structure analysis, the characters and phrases, sentence structure, and meaning of the source program are analyzed and the source program is converted into an intermediate program. In the optimization, the intermediate program is subjected to such processes as the basic block setting, control flow analysis, and data flow analysis. In the resource allocation, to adapt to the instruction sets of the target processor, the variables in the intermediate program are allocated to the register or memory of the target processor. In the code generation, each intermediate instruction in the intermediate program is converted into a program code, and an object program is obtained.

15 [0101]

The generated object program is composed of one or more program codes that cause the computer to execute each step in the flowchart or each procedure of the functional components. There are various types of program codes such as the native code of the processor, and JAVA (TM) byte code. There are also various forms of realizing the steps of the program codes. For example, when each step can be realized by using an external function, the call statements for calling the external functions are used as the program codes. Program codes that realize one step may belong to different object programs. In the RISC processor in which the types of instructions are limited, each step of flowcharts may be realized by combining arithmetic operation instructions, logical operation instructions, branch instructions and the like.

[0102]

After the object program is generated, the programmer activates a linker. The linker allocates the memory spaces to the object programs and the related library programs, and links them together to generate a load module. The generated load module is based on the presumption that it is read by the computer and causes the computer to execute the procedures indicated in the flowcharts and the procedures of the functional components. The program of the present invention can be produced in this way.

#### ■ Use of Program of Present Invention

The program of the present invention can be used as follows.  
[0103]

##### (i) Used as Embedded Program

When the program of the present invention is used as an embedded program, the load module as the program is written into an instruction ROM, together with the Basic Input/Output System (BIOS) program and various pieces of middleware (operation systems). The program of the present invention is used as the control program of the recording/reading apparatus 100 as the instruction ROM is embedded in the control unit and is executed by the CPU.

##### (ii) Used as Application

When the recording/reading apparatus 100 is a hard-disk-embedded model, the Basic Input/Output System (BIOS) program is embedded in an instruction ROM, and various pieces of middleware (operation systems) are preinstalled in the hard disk. Also, a boot ROM for activating the system from the hard disk is provided in the recording/reading apparatus 100.

[0105]

In this case, only the load module is supplied to the

recording/reading apparatus 100 via a transportable recording medium and/or a network, and is installed in the hard disk as one application. This enables the recording/reading apparatus 100 to perform the bootstrapping by the boot ROM to activate an operation system, and then causes the CPU to execute the installed load module as one application so that the program of the present application can be used.

[0106]

As described above, when the recording/reading apparatus 100 is a hard-disk-embedded model, the program of the present invention can be used as one application. Accordingly, it is possible to transfer, lend, or supply, via a network, the program of the present invention separately.

#### <Defect Management Information Processing Unit 13>

The defect management information processing unit 13 described above in the embodiments can be realized as one system LSI.

[0107]

The system LSI is obtained by implementing a bear chip on a high-density substrate and packaging them. The system LSI is also obtained by implementing a plurality of bear chips on a high-density substrate and packaging them, so that the plurality of bear chips have an outer appearance of one LSI (such a system LSI is called a multi-chip module).

The system LSI has a QFP (Quad Flat Package) type and a PGA (Pin Grid Array) type. In the QFP-type system LSI, pins are attached to the four sides of the package. In the PGA-type system LSI, a lot of pins are attached to the entire bottom.

[0108]

These pins function as an interface with other circuits. The system LSI, which is connected with other circuits through such pins as an interface, plays a role as the core of the recording/reading apparatus 100.

5       The bear chip packaged in the system LSI includes a front-end unit, a back-end unit, and a digital processing unit. The front-end unit digitizes analog signals. The back-end unit converts digital data obtained through digital processes into the analog format and outputs the analog data.

10   [0109]

      The internal-structure components shown in the above-described embodiments are implemented in the digital processing unit.

      As described above in "Used as Embedded Program", the load module as the program, the Basic Input/Output System (BIOS) program and various pieces of middleware (operation systems) are written into an instruction ROM. The major improvement of the embodiments is achieved by the load module as the program. It is therefore possible to produce a system LSI of the present invention by packaging the instruction ROM, in which the load module as the program is stored, as the bear chip.

[0110]

      In regards with a specific implementation method, it is preferable to use the SoC implementation or the SiP implementation. 25   The SoC (System on Chip) implementation is a technology for printing a plurality of circuits onto a chip. The SiP (System in Package) implementation is a technology for packaging a plurality of circuits by resin or the like. Through these processes, a system LSI of the present invention can be produced based on the internal



structure of the recording/reading apparatus 100 described in each embodiment above.

[0111]

It should be noted here that although the term LSI is used here, it may be called IC, LSI, super LSI, ultra LSI or the like, depending on the level of integration.

Further, part or all of the components of each recording/reading apparatus may be achieved as one chip. The integrated circuit is not limited to the SoC implementation or the SiP implementation, but may be achieved by a dedicated circuit or a general purpose processor. It is also possible to achieve the integrated circuit by using the FPGA (Field Programmable Gate Array) that can be re-programmed after it is manufactured, or a reconfigurable processor that can reconfigure the connection and settings of the circuit cells inside the LSI. Furthermore, a technology for an integrated circuit that replaces the LSI may appear in the near future as the semiconductor technology improves or branches into another technologies. In that case, the new technology may be incorporated into the integration of the functional blocks constituting the present invention as described above. Such possible technologies include biotechnology.

## Industrial Applicability

[0112]

The optical disc, recording apparatus, and reading apparatus of the present invention can be mass-produced based on the internal structures of them shown in the embodiments above. As such, the optical disc, recording apparatus, and reading apparatus of the present invention has the industrial applicability.